

Raportointijärjestelmän toteuttaminen RESCA-hankkeelle

Tomi Ruusala

Opinnäytetyö
Marraskuu 2014

Ohjelmistotekniikan koulutusohjelma
Tekniikan ja liikenteen ala





Tekijä(t) Ruusala, Tomi	Julkaisun laji Opinnäytetyö	Päivämäärä 18.11.2014
	Sivumäärä 41	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: ()
Työn nimi Raportointijärjestelmän toteuttaminen RESCA-hankkeelle		
Koulutusohjelma Ohjelmistotekniikka		
Työn ohjaaja(t) Ari Rantala		
Toimeksiantaja(t) Oy lamit.fi		
Tiivistelmä <p>Opinnäytetyö toteutettiin Oy lamit.fi:lle, jonka asiakkaana oli Oulun kaupungin RESCA-energiatohokkuushanke. Työn tavoitteena oli saada aikaan raportointijärjestelmä, jolla RESCA voi havaita trendejä energiatohokkuuden muutoksissa. Raportointijärjestelmään liittyen tutkittiin myös eri datan visualisointimahdollisuuksien käytettävyyttä, ominaisuuksia, kustannuksia ja suorituskykyä.</p> <p>Työssä kartoitettiin mahdollisesti soveltuvat piirtokirjastot. Tämän jälkeen karsittiin kirjastojen määrä neljään. Näille valituille kirjastoille suoritettiin tarkempia testejä ja tutkittiin niiden ominaisuuksia. Vertailtuja ominaisuuksia olivat ulkoasu, suorituskyky ja dokumentaatio. Saatujen tulosten pohjalta voitiin valita RESCA-hankkeeseen ja muutama muuhun yleiseen käyttötapaukseen sopivat kirjastot.</p> <p>Raportointijärjestelmä saatiin valmiiksi kaikkien vaatimusten mukaan. Siihen valittiin käytettäväksi Canvas.js-kirjasto, joka oli ainoa testattu kirjasto, joka kykeni piirtämään yli 100 000 datapistettä kuvaajaan. Parhaaksi ilmaiseksi kirjastoksi valittiin Google Charts. Paras maksullinen kirjasto oli Highcharts. Aloittelevalle ohjelmoijalle suositeltiin Canvas.js-kirjastoa.</p> <p>RESCAN raportointijärjestelmä saatiin kaikkien vaatimusten osalta valmiiksi. Järjestelmästä saadaan tuotettua energiatohokkuutta kuvaavia raportteja tiettyjen alueiden rakennuksista.</p>		
Avainsanat (asiasanat) Datan visualisointi, visualisointi, kuvaajat, graafit, energiatohokkuus		
Muut tiedot Liitteitä 5 sivua		



Author(s) Ruusala, Tomi	Type of publication Bachelor's thesis	Date 18.11.2014
		Language of publication: Finnish
	Number of pages 41	Permission for web publication: ()
Title of publication Report generation system for RESCA -project		
Degree programme Software Engineering		
Tutor(s) Rantala, Ari		
Assigned by Oy Iamit.fi		
<p>Abstract</p> <p>The thesis was assigned by Oy Iamit.fi that produced the report generation system for RESCA project. The goal was to create a reporting system capable of producing reports allowing RESCA to detect trends in the changes of building energy efficiency. Concerning the reporting system visualization tools for web-development were to be researched. They were researched for usability, features, costs and performance.</p> <p>In the thesis appropriate drawing libraries were surveyed. After researching all the possible libraries their number was reduced to 4. Further tests and research were conducted on the selected 4 libraries. The properties that were compared were appearance, performance and documentation. With the help of this information the visualization library could be selected for RESCA-project and for a few other use cases.</p> <p>The reporting system was completed according to all requirements. Canvas.js-library was selected to be used for the reporting system, since it was the only library capable of rendering over 100 000 data points. Google charts was selected the best free visualization library. The best paid library was Highcharts. Canvas.js was also the recommended library for a beginner programmer.</p> <p>The reporting system for RESCA was completed with all planned features. The system is capable of producing reports that visualize the energy efficiency of the buildings in the given area.</p>		
Keywords/tags (subjects) Data visualization, visualization, graphs, diagrams, energy efficiency		
Miscellaneous 5 pages of attachments		

SISÄLTÖ

1.	TYÖN LÄHTÖKOHDAT	5
2.	TUTKIMUSASETELMA	6
2.1.	Taustateoria	6
2.2.	Näkökulma	7
2.3.	Tavoitteet.....	8
2.4.	Rajaukset	8
2.5.	Tutkimuskysymykset.....	9
3.	TUTKIMUKSEN TOTEUTUS.....	10
4.	PIIRTOKIRJASTOT	11
4.1.	JavaScript-kirjastot.....	11
4.1.1.	Highcharts	11
4.1.2.	Canvas.js	12
4.1.3.	AMcharts	12
4.1.4.	Google charts	12
4.1.5.	Muut JavaScript-kirjastot.....	12
4.2.	PHP-kirjastot.....	13
4.2.1.	Pchart.....	13
4.2.2.	PHPChart.....	14
4.2.3.	Muut PHP-kirjastot	14
4.3.	Muut kirjastot	15
5.	KIRJASTOJEN VERTAILU	16
5.1.	Yleistä.....	16
5.2.	Ulkoasu.....	16
5.3.	Suorituskyky	18
5.4.	Dokumentaatio	21
5.5.	Valinta	22
6.	STATISTIIKKATIEDOT	25

6.1. Energiajunior	25
6.2. Statistikkatiedon keruu	25
6.3. Statistikkatiedon rajausta	26
7. RAPORTOINTIJÄRJESTELMÄN TOTEUTUS	27
7.1. Piirtoluokat	27
7.1.1. Tietojen käsittely	27
7.1.2. Luokkarakenne	28
7.1.3. Datan ryhmittely	29
7.1.4. Laskenta	31
7.1.5. Piirtoskriptin generointi	31
7.2. Käyttöliittymä	31
7.3. Jatkokehitys	33
8. POHDINTA	35
LÄHTEET	36
LIITTEET	37
Liite 1: MySQL koodi, testidatan generointia varten	37
Liite 2: Statistikkatauluihin tallennettavat tiedot	38
Liite 3: Piirtoluokkien funktiot	39
Liite 4: Bootstrap-multiselect esimerkki	40
Liite 5: Piirtoluokkien käytön koodiesimerkki	40

Kuviot

Kuvio 1. Kuvankaappaus Energiajuniorin raportoinnista	6
Kuvio 2. Google Charts -esimerkkikuvaaja	16
Kuvio 3. Highcharts-esimerkkikuvaaja	17
Kuvio 4. AMcharts-esimerkkikuvaaja	17
Kuvio 5. Canvas.js-esimerkkikuvaaja	18
Kuvio 6. Canvas.js esimerkki dokumentaatiosta	24
Kuvio 7. Tiedon kulku piirtoluokassa	28

Kuvio 8. Piirtoluokkien luokkarakenne	29
Kuvio 9. Kuvakaappaus Oulun rakennuksien lämmöntarpeesta valmistumisvuoden mukaan ryhmitettynä	30
Kuvio 10. Kuvakaappaus raportoinnista, jossa rakennukset on ryhmitetty kunnan mukaan	30
Kuvio 11. Raportointijärjestelmän yläreunan navigointipalkki	31
Kuvio 12. Kuvakaappaus projektien vertailusta	33

Taulukot

TAULUKKO 1. Yhteenveto kirjastojen ulkoasusta	18
TAULUKKO 2. Google Charts suorituskykytestin tulokset	19
TAULUKKO 3. Highcharts suorituskykytestin tulokset	20
TAULUKKO 4. AMcharts suorituskykytestin tulokset	20
TAULUKKO 5. Canvas.js suorituskykytestin tulokset	20
TAULUKKO 6. Dokumentaatioiden arviot	22
TAULUKKO 7. Kirjastojen vertailun yhteenveto	23

Termistö

API

Ohjelmointirajapinta.

Energiajunior

Oy lamit.fi:n web-pohjainen uudisrakennuksien energiatehokkuuden laskenta-ohjelmisto.

Energiaohjelmistot

Energiaohjelmistoilla tarkoitetaan Oy lamit.fi:n kaikkia laskentaohjelmistoja, eli Energiajunior, Energiasenior ja Energiapremier.

GNU GPL

Avoimen lähdekoodin lisenssi, joka antaa kenelle tahansa oikeuden käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmia ja niiden lähdekoodia.

JSFiddle

Web-ympäristö, jossa voi testata JavaScript-, CSS- ja HTML-koodia.

Lämmitystarve

Määrä energiaa, joka rakennukseen täytyy tuoda, jotta sisällä säilyy siedettävä lämpötila.

Standalone-versio

Piirtokirjaston versio, joka ei vaadi mitään muita kirjastoja toimiakseen.

1. TYÖN LÄHTÖKOHDAT

Työn toimeksiantajana toimi Oy Iamit.fi. Se on jyvaskyläläinen energiatehokkuuteen keskittyvä yritys. Yhtiö kehittää erilaisia sovelluksia parantamaan energiatehokkuutta. Raportointijärjestelmän asiakas on Oulun kaupungin RESCA-hanke. RESCA-hankkeen tarkoitus kuvataan hyvin hankkeen etusivulla:

RESCA Oulu on osa valtakunnallista RESCA-hanketta, jonka tavoitteena on lisätä uusituvan energian käyttöä kaupungeissa. Oulun rakennusvalvonnan koordinoima RESCA-hanke keskittyy luomaan valintakonsepteja pientalon erilaisille energialähteille ja niiden optimoiduille yhdistelmille, hybrideille, sekä niihin soveltuville rakennuksen energiatehokkuusratkaisuille. Oulun Hiukkavaaraan rakennetaan uusiutuvan energian pilottialue, josta saadaan hyödyllistä tietoa eri ratkaisuista. (RESCA Oulu 2014)

Työssä tutkittiin erilaisia tapoja visualisoida dataa web-sovelluksissa. Tutkimuksesta selvisi, millaisia kirjastoja ja tekniikoita on käytettävissä web-sovelluksia kehitettäessä. Näihin tekniikoihin ja kirjastoihin keskityttiin eritoten sovelluksen kehittäjän näkökulmasta. Osa kirjastoista on esitelty vain lyhyellä kuvauksella, ja osaa on tutkittu laajemmin RESCA:n raportoinnin tarpeiden pohjalta. Tutkimuksen tuloksena selvisivät parhaat visualisointitekniikat RESCA-hankkeelle ja muutamille muille tilanteille web-kehityksessä.

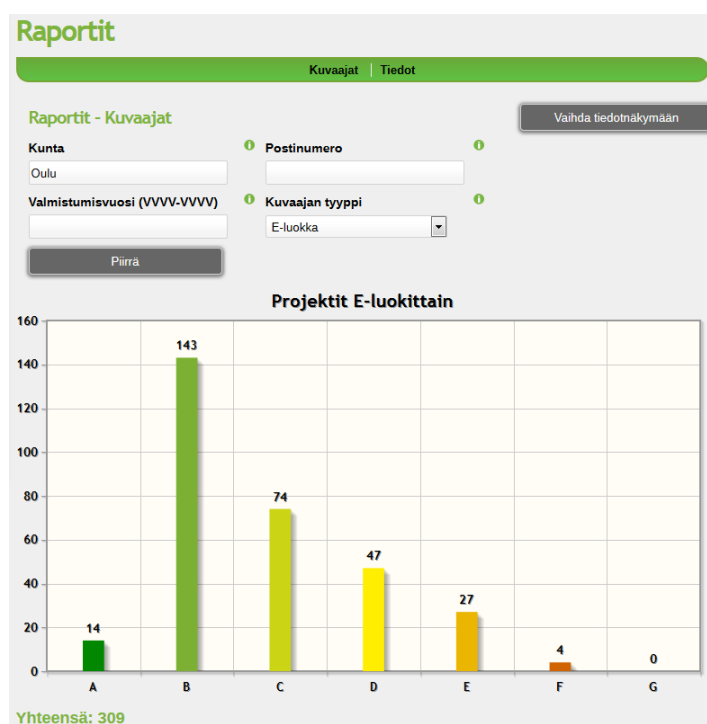
Tarkemmin tutkittuja kirjastoja on tarkasteltu yleiskatsauksen lisäksi suorituskyvyn, dokumentaation ja ulkoasun osalta. Kaikki nämä ovat merkittäviä tekijöitä web-sovellusta kehitettäessä. Kehittäjälle on tärkeätä, että kaikki toiminnot ovat selkeästi kuvattuja. Sovelluksen täytyy olla riittävän kevyt, jotta se toimii kaiken tasoilla laitteilla. Lopuksi myös ulkoasun täytyy olla moderni, jottei käyttäjille anneta vaikutelmaa huonosti tehdystä sovelluksesta.

2. TUTKIMUSASETELMA

2.1. Taustateoria

Oulun RESCA-hankkeesta oltiin alun perin yhteydessä Lamitiin 2013 vuoden alkupuolella. Hanketta varten tarvittiin web-pohjainen energiatehokkuuden raportointijärjestelmä.

Kyseistä järjestelmää alettiin kehittää vuoden 2013 syksyllä. Sen ensimmäinen yksinkertainen versio valmistui saman vuoden lopulla. Se oli Energiajunior-järjestelmän statistiikkatietoihin pohjautuva haku- ja piirtotyökalu. Tästä Energiajuniorin raportoinnista pystyi hakemaan tietoja kunnan, postinumeron ja valmistumisvuoden mukaan kuten kuviossa 1 ilmenee. Haetuilla tiedoilla pystyi piirtämään kuvaajat E-luokkien, lämmitysjärjestelmien ja rakennusten käyttötarkoitusten lukumäärän mukaan. Tämä oli kuitenkin vasta alkua, ja jatkossa tarvittaisiin laajempi raportointijärjestelmä, josta saatavien tietojen avulla voitaisiin havaita tiettyjä trendejä rakennusten energiatehokkuudessa sekä vertailla rakennuksia keskenään.



Kuvio 1. Kuvankaappaus Energiajuniorin raportoinnista

Lisäksi raportointijärjestelmää varten täytyi valita uusi piirtokirjasto. Syksyllä hankittu phpChart oli hieman riittämätön RESCA-hankkeen tarpeisiin nähden. Kuvaajan tallentaminen tiedostoon ei onnistu phpChartilla, joka on kriittinen ominaisuus raportoinnissa. Lisäksi phpChart soveltuu huonosti käytettäväksi suuren datamäärän kanssa ja monimutkaisten kuvaajien piirto on hyvin hankalaa vajaan dokumentaation vuoksi.

Toinen merkittävä osa opinnäytetyötä oli datan muokkaaminen sopivaksi visualisointia varten. Olemassa olevasta Energiajunior-järjestelmästä kerätty statistiikkadata täytyi muokata paikoin muotoon, jossa sitä on helpompi esittää. Esimerkiksi rakennusten sähkönkulutuksista ei voi piirtää omaa pylväskuvaajaa jokaisesta kulutuslukemasta. Kulutuslukemia täytyy ryhmittää esimerkiksi 5000–6000 kWh:a ryhmiin ja niin edelleen. Ryhmien täytyy mukautua eri lukemien lukumäärän ja kulutuslukemien suuruuden mukaan. Esitettävät arvot täytyy valita ja osa arvoista myös laskea.

Energiajunior-järjestelmän statistiikkadatan keruu on alun perin toteutettu 2013 elokuussa. Myöhemmin järjestelmää on muokattu siten, että sinne on lisätty muutamia pieniä tietoja, jotka on voitu laskea ja lisätä jälkikäteen. Lisäksi tietokantaan kerättyä dataa täytyi siistiä. Siistityssä datassa saman rakennuksen tiedoja ei esiinny taulussa kuin yhden kerran, ja siitä on poistettu virheellisesti lisätyt rakennukset.

2.2. Näkökulma

Aihetta lähestyttiin lähinnä RESCA-raportointijärjestelmän tarpeiden mukaan sovelluksen kehittäjän näkökulmasta. Tavoitteena oli löytää tehokas, monipuolinen ja helposti omaksuttava piirtotyökalu, joka täyttää kaikki RESCAN ja muiden Oy Iamit.fi:n projektien tarpeet. Tämän tarkoituksena on säästää aikaa, joka kuluisi piirtotyökalun valintaan ja käytön opetteluun.

2.3. Tavoitteet

Opinnäytteen tavoitteena oli saada aikaan raportointijärjestelmä, jolla RESCA voi havaita trendejä energiatehokkuuden muutoksissa. Raportointijärjestelmään liittyen tutkittiin myös eri datan visualisointimahdollisuuksien käytettävyyttä, ominaisuuksia, kustannuksia ja suorituskykyä.

Lisäksi raportointijärjestelmää varten tehtävästä piirtoluokasta tehtiin yleiskäyttöinen, jolloin luokkaa voidaan käyttää myös muiden projektien kuvaajien piirtoon. Tarkoitus oli luoda piirtoluokka, joka peritään ja muokataan sopivaksi haluttua käyttötarkoitusta varten. Alustavasti on suunniteltu 4 luokkaa (ks. kuvio 8). Chart-luokka sisältää kaikille kuvaajille yhteisiä ominaisuuksia. Tämän luokan apuluokkana käytetään ChartHelpFunctions-luokkaa. Chart-luokka peritään jokaiseen uuteen luokkaan, jotka ovat erityistarkoituserityisluokkia. Aluksi siitä perittyjä luokkia tulee kaksi: BuildingChart ja GeneralChart. Ensimmäinen keskittyy RESCAN raportoinnin kuvaajien piirtoon ja sen erikoisvaatimukseen. GeneralChart-luokan tarkoitus on olla yleinen luokka, jolla voi piirtää mitä tahansa, joka ei vaadi erityistä datan käsittelyä.

Raportointijärjestelmän tulisi olla mahdollisimman dynaaminen, jotta sillä voisi generoida raporteja monipuolisesti käyttäjän tarpeiden mukaan. Raporteista pitäisi selvittää, kuinka energiatehokkuus on kehittynyt tarkastelun kohteena olevalla aikavälillä, sekä millaisia rakennuksia milläkin alueella on. Lisäksi olemassa olevaa tilastotietoa keruuta Energiajunior-järjestelmässä täytyy kehittää vastaamaan paremmin raportointijärjestelmän tarpeita.

2.4. Rajaukset

Tarkemmin tutkittavien kirjastojen määrää täytyi rajata noin viiteen, jotta saataisiin hieman laajemmin tietoa mahdollisesti soveltuvista kirjastoista. Lisäksi resurssien näkökulmasta katsottaessa oli kannattamatonta tutkia täysin soveltumattomien kirjastojen ominaisuuksia tarkasti. Aluksi tutkittavaksi poimittiin noin 10–15 kirjastoa, jotka ovat suosittuja web-kehittäjien keskuudessa. Niistä tutkit-

tiin perustietoja, joiden perusteella valittiin noin 5 kirjastoa tarkempaan testaukseen ja tarkasteluun.

2.5. Tutkimuskysymykset

Keskeisiä ongelmia raportointijärjestelmässä ovat datan muokkaaminen kaikissa tilanteissa sopivaksi visuaalista esitystä varten, sopimattoman datan suodattaminen ja oikeiden asetusten saaminen visualisointityökaluun dynaamisesti. Lisäksi täytyy valita sopiva piirtotyökalu, jolla kuvaajien piirto toteutetaan.

3. TUTKIMUKSEN TOTEUTUS

Aluksi kartoitettiin mahdollisesti soveltuvat piirtokirjastot. Tämän jälkeen karsittiin kirjastojen määrä noin viiteen. Näille valituille kirjastoille suoritettiin tarkempia testejä ja tutkittiin niiden ominaisuuksia. Saatujen tulosten pohjalta voitiin valita RESCA-hankkeeseen sopiva kirjasto.

4. PIIRTOKIRJASTOT

4.1. JavaScript-kirjastot

JavaScript-kirjastot erottuvat selvästi vahvimpina vaihtoehtoina kuvaajien piirtoon RESCA:n raportointia varten joustavuutensa ja laajojen ominaisuuksien vuoksi. JavaScript-kirjastoilla voidaan siirtää piirtämisen aiheuttama prosessori-kuorma palvelimelta käyttäjälle. Piirrettyä kuvaajaa voi muokata piirron jälkeen selaimessa JavaScriptillä. Osalla kirjastoista kuvaajia voi myös zoomata, joka ei ole yksikertaisin toteutuksin mahdollista esimerkiksi PHP-kirjastoilla.

4.1.1. Highcharts

Highcharts on hyvin dokumentoitu monipuolinen JavaScript-piirtokirjasto. Dokumentaatio sisältää mm. JSFiddle-linkit esimerkkeihin kaikista toiminnoista ja aloitusarvoista. Highchartsin yleinen ulkoasu on myös hyvin viimeistelty. Kirjastosta löytyy myös huomattavia erityisominaisuuksia, kuten esimerkiksi erinomainen zoomaus, standalone-versio, joka ei vaadi muita kirjastoja toimiakseen. (Highcharts 2014.)

Highcharts on myös mahdollista ladata omana kokoonpanonaan, jolloin käyttäjä voi itse valita kaikki tarvitsemansa Highchartsin komponentit, kuten esimerkiksi tarvitsemansa kuvaajatyypit. Tällöin web-sovelluksen käyttäjälle lähetettävä JavaScript-tiedosto voidaan pitää mahdollisimman pienenä. (Highcharts 2014.)

Huomattavaa on myös, että osa kirjaston ominaisuuksista vaatii yhtiön Highstock-kirjaston, joka maksaa halvimmillaan yritykselle 600 euroa. Pelkän Highcharts-kirjaston hinta yhdelle web-sovellukselle on kuitenkin vain 300 euroa. (Highcharts 2014.)

4.1.2. Canvas.js

Canvas.js on hyvin dokumentoitu ja selkeärakenteinen JavaScript-kirjasto. Eri-tyisen positiivisena piirteenä dokumentaatiosta voidaan mainita hyvin laajat ja selkeät esimerkit. Kirjasto on vielä todella nuori, sillä sen ensimmäinen versio julkaistiin vasta vuoden 2013 huhtikuussa. Tästä huolimatta kirjasto sisältää laajan skaalan ominaisuuksia. Merkittävimpiä erikoispiirteitä kirjastossa ovat erinomainen suorituskky suurilla datamäärillä, mobiiliystävällisyys ja zoomaus. (Canvas.js 2014.)

4.1.3. AMcharts

AMcharts on JavaScript-piirtokirjasto, joka on varustettu ulkoasueditorilla. Kirjaston Live Editorilla voi luoda halutun ulkoasun ja kopioida koodin haluttuun paikkaan. Kirjaston laajat ominaisuudet ja suhteellisen halpa hinta (110 euroa) tekevät siitä houkuttelevan vaihtoehdon piirtotyökaluksi. (AMcharts 2014.)

4.1.4. Google charts

Google Charts on Googlen JavaScript-piirtokirjasto, jolla kykenee piirtämään lähes kaikkea. Kirjasto sisältää paljon erikoisia ominaisuuksia, kuten datan yhdistämistä SQL-tyylisillä JOIN-komennoilla. Lisäksi kirjasto on täysin ilmainen niin yksityisten kuin yritystenkin käyttöön. (Google Charts 2014.)

4.1.5. Muut JavaScript-kirjastot

Seuraavaksi kuvatut kirjastot ovat JavaScript-kirjastoja, jotka todettiin käyttökelpottomiksi. Niiden lyhyen esittelyn yhteydessä kerrotaan myös syy, miksi kirjaston tarkempi tarkastelu jätettiin tekemättä. Kyseisiä kirjastoja poimittiin mm. Wikipedian piirtokirjastovertailusta (Comparison of JavaScript charting frameworks 2014).

ChartJS on JavaScript-piirtokirjasto, joka ensivaikutelmaltaan vaikuttaa hyvin monipuoliselta ja hyvin dokumentoidulta. ChartJS on myös täysin riippumaton

muista kirjastoista, kuten jQuerystä. Kirjasto päätettiin kuitenkin hylätä, sillä se ei tukenut ns. legend-tietolaatikoita. Lisäksi kirjaston tulevaisuus vaikutti epävaikalta, sillä ChartJS on yhden henkilön kehittämä.

Raphaël on JavaScript-vektorigrafiikkakirjasto, jolla voi piirtää lähes mitä tahansa. Raphaël taipuu hyvin piirtämään mitä monimutkaisimpia kuvioita, mutta sen toiminta todettiin liian vaikeaselkoiseksi kuvaajien piirtoon. Kirjaston API on myös liian monimutkainen ja vaikuttaa siltä, ettei kirjastoa käyttäen pysty piirtämään kuvaajia, joissa on paljon datapisteitä. Todennäköisesti kuvaajien piirtoon keskittynyt kirjasto tarjoaa laajemmat ominaisuudet ja paremman ohjelmointirajapinnan.

Flotr on hyvin vajaa ominaisuuksien osalta ja kaikin puolin RESCA:n tarpeisiin nähden riittämätön. Kirjasto vaatii myös Prototype-kirjaston toimiakseen, jota Lamit ei halua ottaa käyttöön. Dokumentaatio on suppea ja esimerkkejä on tarjolla hyvin rajallinen määrä. Tästä voisi mahdollisesti aiheutua runsaasti lisää työtä projektin edetessä. Lisäksi kirjaston kehitys vaikuttaa pysähtyneen.

JqPlot hylättiin heti tarkastelun alussa pysähtyneen kehityksen vuoksi. Kehitys on pysähtynyt 2013, eikä siten ole oletettavasti tämän hetken standardien tasoa. Lisäksi dokumentaatio oli keskivertoa selvästi huonompi.

4.2. PHP-kirjastot

PHP-kirjastot eivät sovellu RESCA-projektiin, eivätkä muihin Lamitin projekteihin, sillä kuvaajaa täytyy pystyä tarpeen vaatiessa muokkaamaan reaaliajassa. Lisäksi PHP-kirjastoilla ei ole mahdollista zoomata kuvaajia.

4.2.1. Pchart

Pchart on PHP:n GD- ja FreeType-laajennoksiin perustuva piirtokirjasto (Pchart 2014). Kirjasto vaikuttaa ammattimaisesti tehdyltä ja kohtuullisen hyvin dokumentoidulta. Kirjasto on jaossa GNU GPLv3 -lisenssin alaisena ja lisäksi ostettavissa kaupalliseen käyttöön 390 euron hintaan (Pchart 2014). Kirjasto sisältää

valtavan skaalan ominaisuuksia ja niiden viidakoon voikin hukkua koodatessa. Kirjasto vaikuttaa myös kotisivujen esimerkkien perusteella melko työläältä käyttää. Yksinkertaisetkin kuvaajat vaativat hurjia määriä koodia vastaaviin JavaScript-kirjastoihin verrattuna.

4.2.2. PHPChart

PHPChart on PHP-kirjasto, joka luo JavaScriptillä kuvaajat käyttäen JqPlotia (PhpChart 2014). PHPChartilla saa helposti jo muutamalla rivillä PHP-koodia aikaan yksinkertaisia kuvaajia. Kirjaston teknisestä tuesta saa kysymyksiin vastauksen parissa arkipäivässä. Monimutkaisempia kuvaajia piirrettäessä käyttö kuitenkin muuttuu hankalammaksi. Puutteellinen dokumentaatio tekee monista pienistä muutoksista vaikeita. Dokumentaatio on sekavaa ja puutteellista. Lisäksi ohessa joutuu selaamaan JqPlotin dokumentaatiota, jotta löytäisi kaikki haluamansa komennot. Kirjasto on riippuvainen jQuerystä ja tietysti JqPlotista. PHPChartin sivuilla ei myöskään ole kunnollisia versiotietoja, eikä tietoa kirjaston tulevaisuudesta.

4.2.3. Muut PHP-kirjastot

JpGraph on PHP-kirjasto kohtuullisen hyvällä dokumentaatiolla. Kuvaajat vaikuttavat ulkonäöltään vanhanaikaisilta. Kehitys on kuitenkin lopetettu jo 2010, joten tarkempi tarkastelu jätettiin tekemättä (JpGraph 2014).

Libchart on PHP-kirjasto, joka on tarjolla ainoastaan GNU GPL lisenssin alla. Ominaisuudet vaikuttavat hyvin suppeilta kilpailijoiden ominaisuuksiin nähden. Dokumentaatio on kuitenkin tehty hyvin. Kirjaston kehitys on pysähtynyt jo 2011.

Edellä mainittujen valmiiden kirjastojen lisäksi kuvaajia voisi piirtää myös käyttäen PHP:n GD-kirjastoa, johon suurin osa kaikista PHP:n piirtokirjastoista pohjautuu. Se ei tosin ole kovin kannattavaa, sillä olisi hyvin aikaa vievää lähteä kehittämään esimerkiksi Pcharttia vastaavaa kirjastoa itse. (Image Processing and GD 2014)

4.3. Muut kirjastot

PHP sisältää GD-kirjaston, joka on alemman tason piirtokirjasto. Se sisältää peruspiirto-ominaisuuksia esimerkiksi ympyrän, neliön ja kirjoittamisen. (Image Processing and GD 2014.)

Tällä tavalla piirtotyökalun tekeminen on kuitenkin työlästä, eikä millään tapaa suositeltavaa, koska olemassa on paljon hyviä valmiita kirjastoja, jotka pohjautuvat GD-kirjastoon.

Lisäksi on olemassa kirjastoja, jotka piirtävät kuvaajat Microsoft Silverlightilla, kuten **Visifire** (Visifire 2014). Visifire vaatii Microsoftin palvelimen, joten Linux-ympäristössä kyseinen tekniikka ei ole vaihtoehto.

Flashilla piirto onnistuu myös esimerkiksi "**PHP/SWF Charts**"-kirjastolla. Kyseessä on PHP-kirjasto, joka käyttää Flashiä piirtämiseen (PHP/SWF Charts 2014). Flash on myös poissuljettu vaihtoehto, sillä sitä ei haluta ottaa käyttöön Lami:lla.

5. KIRJASTOJEN VERTAILU

5.1. Yleistä

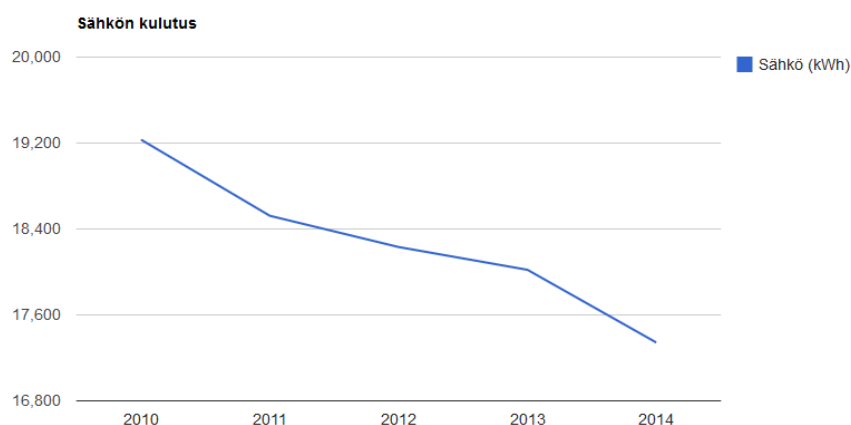
Visualisointityökalujen arvioinnissa jokainen työkalu arvostellaan asteikolla 0-5 jokaisen ominaisuuden osalta, jotka määritellään myöhemmin tässä luvussa. Arvioitujen ominaisuuksien perusteella muodostetaan lopputulos, jolla voidaan valita tarpeeseen sopiva visualisointityökalu. RESCA-hankkeen lisäksi ominaisuuksien perustella annetaan suositus sopivimmasta työkalusta muutamille käyttötapauksille: paras ilmainen työkalu, paras maksullinen työkalu ja työkalu yksinkertaisten kuvaajien piirtoon.

5.2. Ulkoasu

Ulkoasua vertaillaan piirtämällä vakioasetuksilla yksinkertainen viivadiagrammi, jonka ulkonäköä arvioidaan asteikolla 0-5. Arvioinnissa tarkastellaan, onko kuvaajan ulkoasussa virheitä ja kuinka ammattimaiselta ulkoasu vaikuttaa. Sen lisäksi tutkitaan, vaikuttavatko fontit ja yleinen ilme nykyaikaiselta.

Google Charts

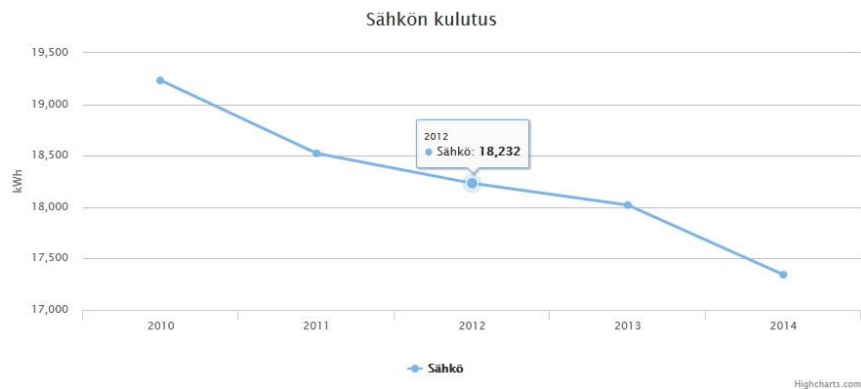
Ensi silmäyksellä kuvaaja vaikuttaa hyvältä. Virheitä ei ole havaittavissa. Fontti ja sen koko on hyvä. Ulkoasu on kuitenkin yleisilmeeltään hieman tylsä (ks. kuvio 2). Arvosana: 4,5/5.



Kuvio 2. Google Charts -esimerkkikuvaaja

Highcharts

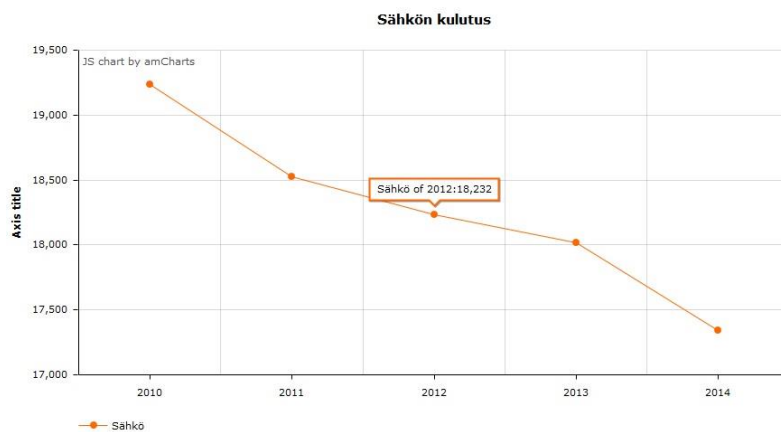
Datapistekyltit antavat heti todella hyvän vaikutelman kuvaajasta. Fontti on selkeä. Kaikki kuvaajassa huokuu ammattimaista laatua (ks. kuvio 3). Arvosana: 5/5.



Kuvio 3. Highcharts-esimerkkikuvaaja

AMcharts

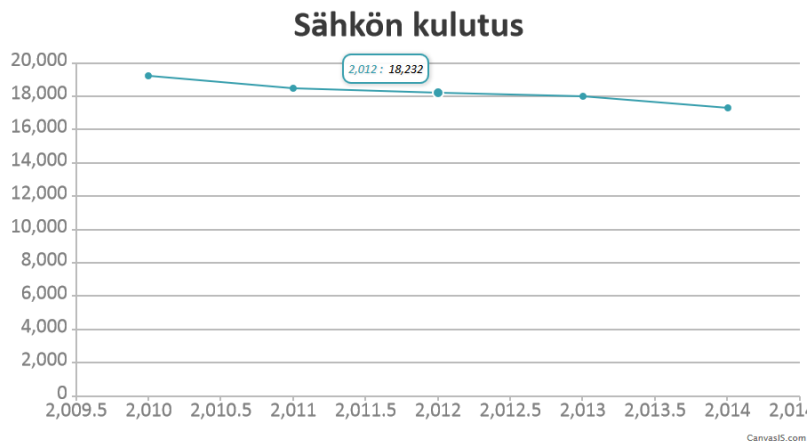
Ulkoasu on hyvin tyylikäs kaikin puolin. Mikään kuvaajassa ei pistä erityisesti silmään (ks. kuvio 4). Arvosana: 5/5.



Kuvio 4. AMcharts-esimerkkikuvaaja

Canvas.js

Ulkoasussa pistää silmään oikean alalaidan x-akselin viimeisen luvun leikkautuminen. Fontti vaikuttaa suurehkolta ja hieman suttuiselta. Kuvaaja on myös puuristunut ylälaitaan, sillä y-akselin asteikko on skaalautunut huonosti (ks. kuvio 5). Arvosana 3/5.



Kuvio 5. Canvas.js-esimerkkikuvaaja

Yhteenveto

Perusulkoasun perusteella kaikki paitsi Canvas.js ovat hyvin näyttäviä, ja soveltuvat hyvin ammattimaiseen käyttöön (ks. taulukko 1). Toki myös Canvas.js:n ulkoasua voi itse muokata halutunlaiseksi. Oikean alalaidan leikkautumiselle ei voi mitään, sillä kyseiseen ongelmaan ei Canvas.js:n kehittäjien mukaan ole toistaiseksi ratkaisua.

TAULUKKO 1. Yhteenveto kirjastojen ulkoasusta

Kirjasto	Google Charts	Highcharts	AMcharts	Canvas.js
Arvosana	4,5/5	5/5	5/5	3/5

5.3. Suorituskyky

Suorituskykyä arvioitiin mittaamalla aika, joka selaimelta kuluu kuvaajan renderöintiin. Kaikki suoritusajat on mitattu samalla kannettavalla työtietokoneella Firefoxin versiolla 30.0. Todellisen mittausdatan kaltaista testidataa on ennen testejä lisätty tietokantaan 200 000 riviä. Tästä datasta haetaan testiin vaadittu määrä rivejä. Data haetaan asynkronisesti AJAX:lla JavaScriptiin JSON-tiedostona. Haettavat rivit sisältävät UNIX-aikaleiman ja mittarilukeman. Lukemat on muodostettu kuvaamaan oikean kaltaista kulutus dataa, esim. sähkön kulutus. Mittarilukema on numerosta 1 alkava float-luku, johon lisätään MySQL-satunnaisluku + 1 joka kierros. Täten saadaan kulutuslukemia, jotka kasvavat

aivan kuten sähkönkulutus. Lukemat on muodostettu käyttäen liitteestä 1 löytyvää MySQL proseduuria.

Suoritus aika on mitattu jokaisella kirjastolla kolme kertaa. Saaduista kolmesta tuloksesta lasketaan kirjastokohtainen keskiarvosuoritus aika. Suoritettava testisivu on ladattu ennakoon, jotta kaikki JavaScriptit ja muut resurssit löytyvät paikallisesti. Testi suoritetaan painamalla selaimen päivitä-painiketta ja tarkastamalla selaimen konsolista kulunut aika.

Testissä 1 kuvaajat on piirretty 5 datapisteellä, testissä 2: 10 000 datapisteellä ja testissä 3: 100 000 datapisteellä. Testissä 1 on käytetty kovakoodattua dataa. Testeissä 2 ja 3 käytetään yllä mainittua tietokantadataa. Lisäksi kirjaston yleistä suorituskäytöntuntemaa arvioidaan asteikolla 0-5. 0 tarkoittaa täysin käytökelvotonta ja 5 tarkoittaa, ettei suorituskäytössä ole mitään moitittavaa. Tuntemuksen arviointiin vaikuttaa muun muassa piirretyn kuvaajan datapisteiden välillä hiiren liikuttamisen aiheuttama viive, inforuutujen avautumisnopeus ja muu pätkeminen tai jäätyminen kuvaajia käsitellessä.

JavaScript-kirjastojen suoritus aika mitattiin käyttäen seuraavanlaista koodia:

```
console.time("Piirto aika");  
Suoritettava piirtokoodi  
console.timeEnd("Piirto aika");
```

Google Chart

Google Chart suoriutuu hyvin ensimmäisestä testistä, mutta toiminta alkaa takkuilla pahasti jo testissä 2. Kolmas testi keskeytyi (ks. taulukko 2).

TAULUKKO 2. Google Charts suorituskäytötestin tulokset

	Suoritus aika	Suoritus aika	Suoritus aika	Keskiarvo	Tuntuma
Testi 1	99,32ms	99,21ms	95,99ms	98ms	5/5
Testi 2	1912,33ms	2020,21ms	1918,55ms	1950ms	2,5/5
Testi 3	Keskeytyi	Keskeytyi	Keskeytyi	-	Keskeytyi

Highcharts

Testissä 3 Highcharts kykenee piirtämään datapisteet, mutta käyttötuntuma on todella tökerö (ks. taulukko 3).

TAULUKKO 3. Highcharts suorituskykytestin tulokset

	Suoritus aika	Suoritus aika	Suoritus aika	Keskiarvo	Tuntuma
Testi 1	47,81ms	47,24ms	46,89ms	47ms	5/5
Testi 2	167,2ms	174,08ms	195,01ms	179ms	2,5/5
Testi 3	889,64ms	1027,8ms	1006,14ms	975ms	0,5/5

AMcharts

AMchartsin testissä 1 suoritus aika on niin lyhyt, että ajoitus ei välttämättä toiminut oikein (ks. taulukko 4).

TAULUKKO 4. AMcharts suorituskykytestin tulokset

	Suoritus aika	Suoritus aika	Suoritus aika	Keskiarvo	Tuntuma
Testi 1	2,55ms	2,48ms	2,44ms	2,49ms	5/5
Testi 2	1690,5ms	1593,57ms	1784,47ms	1690ms	2/5
Testi 3	Keskeytyi	Keskeytyi	Keskeytyi	-	Keskeytyi

Canvas.js

Canvas.js suoriutuu mainiosti kaikista testeistä. Renderöintiajat ovat selkeästi parempia kuin muilla testatuilla kirjastoilla (ks. taulukko 5).

TAULUKKO 5. Canvas.js suorituskykytestin tulokset

	Suoritus aika	Suoritus aika	Suoritus aika	Keskiarvo	Tuntuma
Testi 1	27,71ms	29,19ms	28,69ms	28,5ms	5/5
Testi 2	71,33ms	72,06ms	70,97ms	71,5ms	5/5
Testi 3	188,39ms	191,38ms	192,2ms	191ms	4,5/5

Yhteenveto

Ensimmäinen testi sujui loistavasti kaikilta testatuilta kirjastoilta. Kaikki suoriutuivat renderöinnistä nopeasti ja käyttötuntuma oli kaikilla kirjastoilla hyvä (ks. taulukko 6).

Toisessa testissä havaittiin jo selkeitä merkkejä siitä, että Canvas.js olisi suorituskkyisin kirjastoista. Google Charts ja AMcharts renderöivät lähes 2 sekuntia ja käyttötuntuma oli tyydyttävä. Highcharts taas renderöi kuvaajan hyvään aikaan, mutta käyttötuntuma oli heikko. Siirrettäessä osoitinta kuvaajan päällä oli havaittavissa merkittävää nykimistä, joka vaikutti käyttökokemukseen hyvin negatiivisesti. Canvas.js puolestaan renderöi kuvaajan hyvin nopeasti ja käyttötuntuma oli moitteeton (ks. taulukko 5).

Viimeisessä suorituskkytestissä voittaja oli selvä. Canvas.js renderöi kaikki 100 000 datapistettä alle 200 millisekunnin ja tuntuma oli lähes virheetön. Highcharts kykeni renderöimään datapisteet, mutta jumitti lähes koko selaimen. Amcharts ja Google Charts puolestaan eivät saaneet kuvaajaa renderöityä ollenkaan.

Testeissä osoittautui, että suorituskkyvyn näkökulmasta vaihtoehtoja oli vain yksi, Canvas.js. Dokumentaation mukaan myös Highchartsin pitäisi kyetä piirtämään suuriakin datapistemääriä Highstock-lisäkirjaston kanssa. Näissä testeissä kyseinen lisäkirjasto ei kuitenkaan ollut mukana, jolloin se ei kykene piirtämään 100 000 datapistettä moitteetta.

5.4. Dokumentaatio

Dokumentaatiota arvioidaan sen kattavuuden, selkeyden, esimerkkien ja live-demojen perusteella. Arviointiasteikkona käytetään asteikkoa 0-5. 0 tarkoittaa, että dokumentaatiota ei löydy ja 5 tarkoittaa, että dokumentaatio kattaa kaikki kirjaston funktiot ja vakioarvot, sekä kaikki mahdolliset arvot funktioille on dokumentoitu erikoistilanteineen. Lisäksi esimerkkejä funktioiden käytöstä löytyy kattavasti. Dokumentaatiosta tulee löytyä myös muokattavat live-demot ominaisuuksien kokeilua varten.

Dokumentaation arvioinnit on koottu taulukkoon 6, josta ilmenee, että Highchartsilla on paras dokumentaatio ja puolestaan Google Chartsilla huonoin.

TAULUKKO 6. Dokumentaatioiden arviot

Kirjaston nimi	Arvio	Huomiot
Google Charts	3/5	Ensivaikutelma on sekava. Kaikista funktioista ei ole esimerkkejä. Halutun toiminnon löytäminen tuntuu hyvin työläältä välillä. Live-demoja ei ole.
Highcharts	5/5	Dokumentaatioissa on hyvä puurakenne, joka on jaettu loogisesti. Halutut toiminnot löytyvät ripeästi. Jokaisesta arvosta, jonka Highchartsin voi syöttää, löytyy JSFiddle-linkki.
AMcharts	3,5/5	Dokumentaatio on rakenteeltaan erilainen kuin muilla kirjastoilla, joka vaikeuttaa sen käyttöä. Live-demoja ei ole ja kaikista toiminnoista ei ole edes esimerkkejä.
Canvas.js	4,5/5	Dokumentaatiosta löytyy runsaasti esimerkkejä. Kaikki toiminnot on dokumentoitu selkeään puurakenteeseen ja jokaisesta toiminnosta löytyy live-demat.

5.5. Valinta

Syksyllä 2013 Lamitille hankittiin kuvaajien piirtoa varten PHP/JavaScript-kirjasto, PhpChart. Se osoittautui kuitenkin riittämättömäksi tarpeiden kasvaessa. Näihin kasvaviin tarpeisiin kuului muun muassa tuki suurelle datamäärälle (yli 100 000 datapistettä). Lisäksi toivottiin, että kuvaajaa voi muokata piirtämisen jälkeen selaimessa. Tämä onnistuikin kaikilla JavaScript-kirjastoilla. Suuren datamäärän tarve johtuu toisesta Lamitor-nimisestä mittausdataprojektista, joka otetaan myös huomioon kirjastoa valittaessa, jotta vältetään useamman kirjaston hankinnalta ja käytöltä.

Tulukon 7 kootuista tuloksista valittiin parhaat kirjastot seuraaville käyttötarkoituksille: RESCAN raportointi, paras ilmainen työkalu, paras maksullinen työkalu ja työkalu aloittelevalle ohjelmoijalle.

TAULUKKO 7. Kirjastojen vertailun yhteenveto

	Google Charts	Highcharts	AMcharts	Canvas.js
Ulkoasu	4,5/5	5/5	5/5	3/5
Suorituskyky (testi 3)	Keskeytyi	975ms	Keskeytyi	191ms
Käyttötuntuma	Keskeytyi	0,5/5	Keskeytyi	4,5/5
Dokumentaatio	3/5	5/5	3,5/5	4,5/5

RESCAN raportointiin valittiin Canvas.js, koska kirjasto oli ainoa joka kykeni piirtämään ja toimimaan riittävän hyvin 100 000 datapisteellä (ks. taulukko 7). Se on myös hyvin dokumentoitu ja tukee zoomausta. Highcharts oli toinen potentiaalinen vaihtoehto, mutta 100 000 pisteen piirtoon vaaditaan lisäpaketti Highstock, joka maksaisi 577 euroa lisää. Tällöin kirjaston hinta nousisi jo lähes 900 euroon, jonka vuoksi päädyttiin Canvas.js-kirjastoon.

Paras täysin ilmainen työkalu on selkeästi Google Charts. Se yltää lähes samaan tulokseen kuin 3 muuta tarkemmin tutkittua työkalua. Jos kirjaston käyttäjällä ei ole suuria erikoistarpeita, Google Charts on ehdottomasti suositeltavin työkalu.

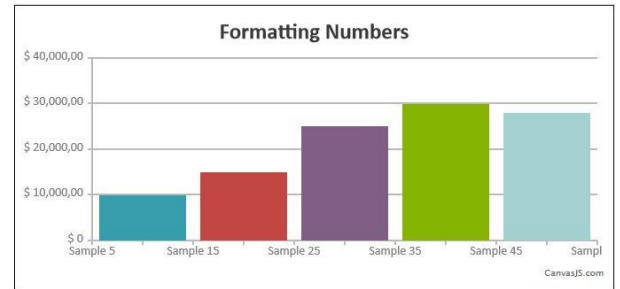
Paras maksullinen työkalu on Highcharts. Highcharts on erittäin näyttävä jo vakioasetuksillakin. Lisäksi sillä on ehdottomasti paras dokumentaatio kaikista kirjastoista. Suureen datamäärään se ei tosin kykene vastaamaan ilman Highstock-lisäpakettia, mutta muuten Highcharts loistaa kaikessa, mitä se tekee. 300 euron hintaan se on ehdoton ykkösvalinta.

Aloittelevalle ohjelmoijalle suositeltavin kirjasto on Canvas.js sen erinomaisten muokattavien esimerkkien vuoksi. Jokaisella dokumentaation sivulla on koodiesimerkki, jota voi muokata halutunlaiseksi, ja muokkauksen vaikutukset voidaan nähdä heti sivun oikeassa laidassa olevasta kuvaajasta (ks. kuvio 6).

```

Try it Yourself by Editing the Code below.
<!DOCTYPE HTML>
<html>
<head>
<script type="text/javascript">
window.onload = function () {
    var chart = new CanvasJS.Chart("chartContainer", {
        title: {
            text: "Formatting Numbers"
        },
        axisY: {
            valueFormatString: "$ #,###,##0", //try properties here
        },
        axisX: {
            valueFormatString: "Sample #"
        },
        data: [
            {
                type: "column",
                dataPoints: [
                    { x: 10, y: 10000000 },
                    { x: 20, y: 15000000 },
                    { x: 25, y: 25000000 },
                    { x: 35, y: 30000000 },
                    { x: 45, y: 28000000 },
                    { x: 50, y: 25000000 }
                ]
            }
        ]
    });
    chart.render();
}

```



Kuvio 6. Canvas.js esimerkki dokumentaatiosta

6. STATISTIIKKATIEDOT

6.1. Energiajunior

Energiajunior on yksi Lamitin energiaohjelmistoista. Se on myös tuoteperheen käytetyin ohjelma. Kaksi muuta perheeseen kuuluvaa ohjelmaa ovat Energiase-nior ja Energiapremier. Energiaohjelmistoilla simuloidaan rakennusten energia-tehokkuutta, eli yksinkertaistettuna lasketaan rakennukseen tulevat, ja sieltä poistuvat energiat. Sovellusten päätarkoitus on laskea energiatodistuksissa tar-vittavat laskut. (Energiajunior 2014)

Tuoteperheen kolme ohjelmaa eroavat käyttötarkoituksiltaan vain hieman. Energiajunior on tarkoitettu uudisrakennusten energiatodistusten laadintaan, kun taas Energiasenior on jo olemassa olevia rakennuksia varten kehitetty ver-sio samasta ohjelmasta. Uusin tuotteista, Energiapremier on jäähdytettyjen ra-kennusten laskentaan kehitetty laskentasovellus. Niiden kaikkien takana toimii sama laskentaydin.

Kaikki tämän työn raportoinnissa käytettävä statistiikka on peräisin Energiaju-nior-, ja Energiasenior-järjestelmistä. Statistiikkaa on kertynyt noin 2500 projek-tista 2013 kesältä lähtien, jolloin statistiikkaa ryhdyttiin ensimmäistä kertaa ke-räämään.

6.2. Statistiikkatiedon keruu

Statistiikkaan kerätään liitteessä 2 mainittavat tiedot valmiista energiaohjelmistojen projekteista. Tiedot sisältävät perustietoja rakennuksesta, kuten lämmitysjärjestelmät, valmistumisvuoden ja pinta-alan. Lisäksi statistiikkaan lisätään laskettuja arvoja kuten lämpökuormat, jotka ovat (liitteessä 2) qh-alkuiset tietokannan sarakkeet.

Alkuperäinen statistiikan keruu on aloitettu vuoden 2013 elokuussa. Tämän jälkeen siihen on lisätty vain muutamia pieniä tietoja, jotka liittyvät lähinnä

helpompien hakujen tekemiseen statistiikasta. Jälkikäteen lisättyjä tietoja ovat ainakin käyttäjän id ja kunta. Lisäksi joidenkin tietojen tallennuksessa on ollut virheitä ja kyseiset tiedot on täytetty uudelleen tietokantakyselyjen avulla. Lisäksi osa projektien statistiikasta laskettiin kokonaan uudestaan bugien aiheuttamien virheiden vuoksi.

6.3. Statistiikkatiedon raja

Statistiikkaan tallennettavia projekteja valvotaan ja rajoitetaan, jotta tilastot pysyisivät siisteinä, eikä mukaan pääsisi virheellisiä projekteja. Niitä valvotaan kolmella eri tavalla: statistiikkaan lisätään ainakin toistaiseksi vain projektit, joista käyttäjä on ladannut energiatodistuksen. Energiatodistusta ladattaessa tarkastetaan, onko käyttäjä merkitty testikäyttäjäksi. Mikäli kyseessä on testikäyttäjä, statistiikkaa ei tallenneta. Viimeiseksi tarkastellaan rakennuksen tietoja. Näin varmistetaan, että ne eivät poikkea annetuista rajoista. Tarkastettavia arvoja ovat mm. valmistumisvuosi, pinta-ala, auringon lämpökuorma ja projektin tyyppi.

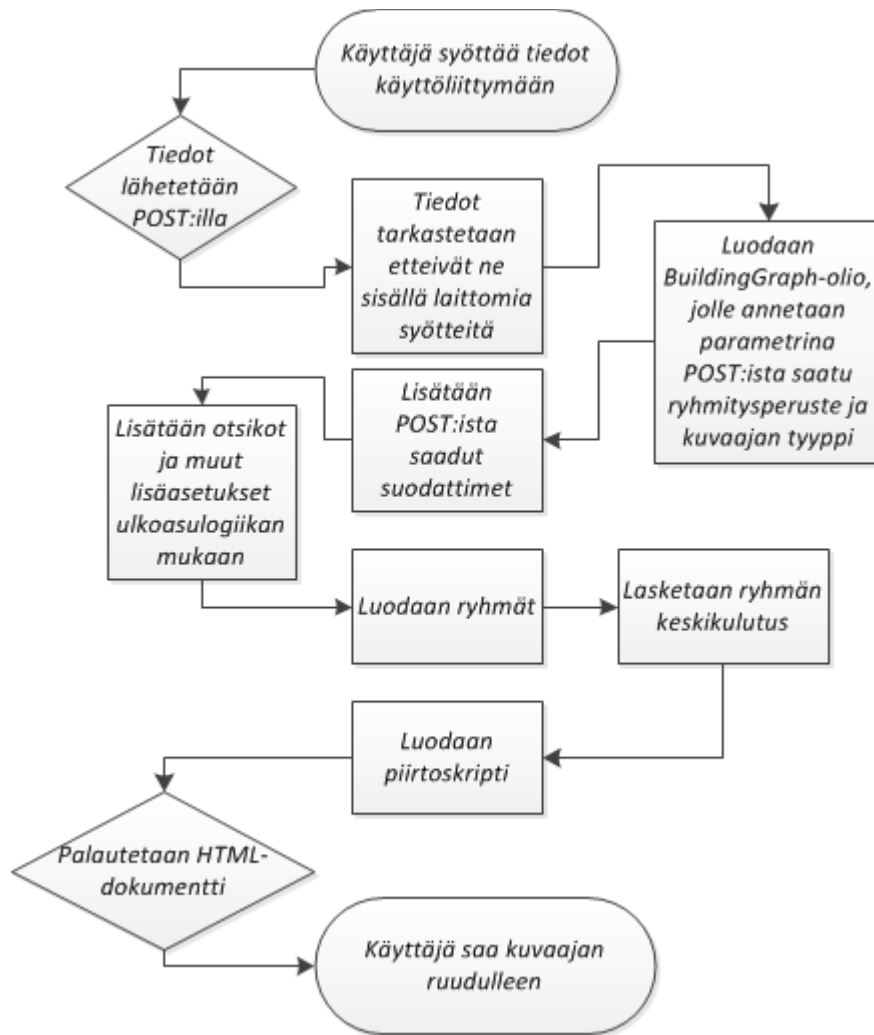
7. RAPORTOINTIJÄRJESTELMÄN TOTEUTUS

7.1. Piirtoluokat

7.1.1. Tietojen käsittely

Raportoinnin tiedot kulkevat käyttäjän selaimelta POST-metodin välityksellä palvelimelle, jossa ne ensin tarkastetaan laittomien syötteiden varalta. Tarkastuksen läpäistyään tiedot syötetään BuildingChart-oliolle. Tiedoista generoidaan JavaScript-koodi, jolla kuvaaja piirretään. Generoitu koodi lähetetään muun HTML-koodin mukana käyttäjän selaimeen, jossa siitä piirretään kuvaaja.

Piirtäminen vastaanotetuista rakennuksen tiedoista aloitetaan luomalla BuildingChart-olio (ks. kuvio 7). BuildingChart-luokka ottaa vastaan parametrina ryhmitysperusteen ja kuvaajan tyypin (ks. liite 3). Tämän jälkeen oliolle voidaan syöttää suodattimia mm. addFilter-metodilla. Erityiskäsittelyä vaativille arvoille kuten lämmitysjärjestelmille ja rakennustypeille on omat erityisfunktionsa.



Kuvio 7. Tiedon kulku piirtoluokassa

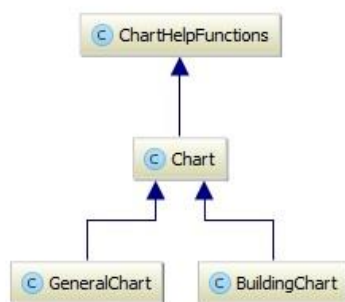
Muita asetuksia voidaan myös asettaa tässä vaiheessa mm. settereillä, jotka löytyvät liitteestä 3. Kun kaikki halutut asetukset on käytössä, voidaan kutsua `createGroupedBuildingsPerItem`-metodia tai `createGroupedBuildingsPerArea`-metodia, jotka luovat ryhmät. Ensimmäinen näistä on tarkoitettu kappalemäärin laskemiseen ja jälkimmäinen yksikkötiheyden laskentaan. Toistaiseksi yksiköitä on valittavissa lämmöntarve ja sähkön kulutus. Kun ryhmitys on tehty, jäljellä on `drawGraph`-metodin kutsuminen, joka generoi ja palauttaa `Canvas.js:n` JavaScript-koodin.

7.1.2. Luokkarakenne

Piirtoluokkien luokkarakenne on yksinkertainen. `Chart`-luokka perii `ChartHelp-Functions`-luokan. Kaikkiin erikoistarkoituksiin tehdään omat luokat, jotka peri-

vät Chart-luokan (ks. kuvio 8). Erikoistarkoituksiin on toistaiseksi vain BuildingChart- ja GeneralChart-luokat.

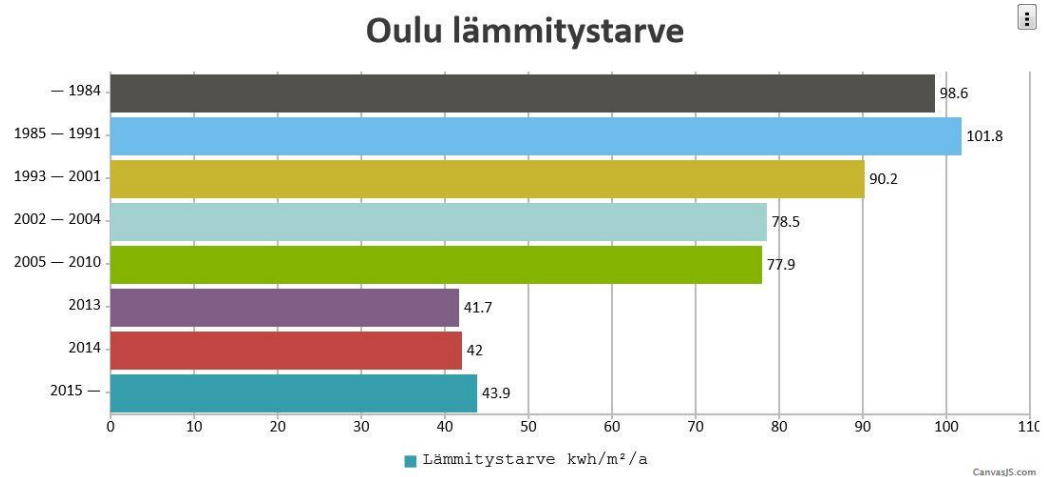
GeneralChart on yleinen piirtoluokka, jolla voidaan piirtää kuvaajia datasta, joka ei vaadi laskentaa tai muuta erityistä muokkaamista. BuildingChart on erikoisluokka RESCA:n raportointia varten, jolla voidaan mm. ryhmittää rakennuksia dynaamisesti n-määrään ryhmiä. Chart-luokka puolestaan sisältää kaikki yleiset funktiot, joita tarvitaan kaikissa erikoisluokissa.



Kuvio 8. Piirtoluokkien luokkarakenne

7.1.3. Datan ryhmittely

Dataa voidaan ryhmittää minkä tahansa tietokentän perusteella syöttämällä BuildingChart-luokan konstruktoriin group_by-parametriin halutun tietokentän nimi (ks. liite 3). Ryhmät muodostetaan jakamalla kaikki rakennukset suurin piirtein samankokoisiin ryhmiin. Ryhmäkoot voivat kuitenkin poiketa huomattavasti oletuskoosta, jos yhteen ryhmään kuuluvia rakennuksia on paljon. Esimerkiksi, edellä olevasta kuviosta nähdään, kuinka vuosina 2013 ja 2014 valmistuneita rakennuksia on enemmän kuin ryhmässä kuuluisi olla, jolloin ne saavat omat ryhmänsä (ks. kuvio 9).



Kuvio 9. Kuvakaappaus Oulun rakennuksien lämmöntarpeesta valmistumisvuoden mukaan ryhmitettynä

Dataa voidaan myös ryhmittää siten, että kaikki eri arvon sisältävät rakennukset saavat oman ryhmänsä. Näin tehdään muun muassa kunnan mukaan ryhmitetyssä (ks. kuvio 10).

Kuvaajat

Kunnat i

Espoo x Helsinki x Jyväskylä x Kuopio x Lahti x Oulu x Tampere x Turku x Vaasa x Vantaa x

Kuvaajan tyyppi i **Ryhmitysperuste** i **Ryhmien määrä** i

Lämmitystarve Kunta 10

Rajaukset

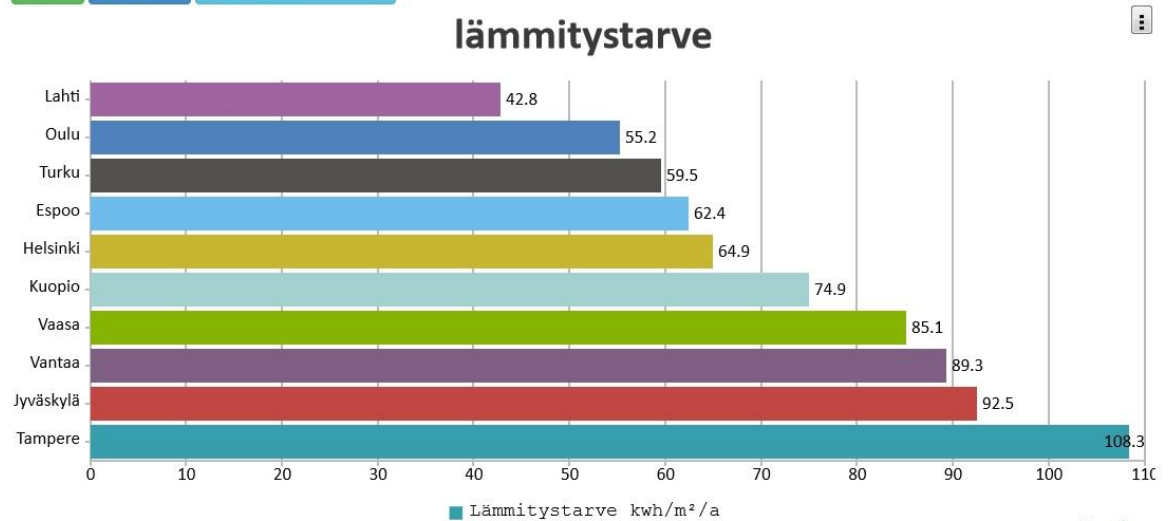
Rakennusvuosi

Alkaen VVVV Päättymen VVVV Rakennus Kaikki Lämmitysjärjestelmät 17 valittuna

Rakennustyyppi Valitse: Kaikki | Ei mitään

☒ Pien-, rivi- ja ketjutalo ☒ Asuinkerrostalo ☒ Toimistorakennus ☒ Liikerakennus ☒ Majoitusliikerakennus ☒ Opetusrakennus ja päiväkotito ☒ Liikuntahalli ☒ Sairaala

Hae Tallenna Tallennetut suodattimet



Rakennusten lukumäärä

Espoo: 175 Helsinki: 134 Jyväskylä: 275 Kuopio: 17 Lahti: 10 Oulu: 326 Tampere: 111 Turku: 18 Vaasa: 21 Vantaa: 251

Yhteensä: 1338

Kuvio 10. Kuvakaappaus raportoinnista, jossa rakennukset on ryhmitetty kunnan mukaan

7.1.4. Laskenta

Laskentaa raportointityökalussa tehdään toistaiseksi vielä hyvin vähän. Tällä hetkellä työkalulla saa laskettua ryhmälle sähkönkulutuksen tai lämmitystarpeen kilowattitunteina neliötä kohti vuodessa. Lisäksi voidaan laskea rakennusten kappalemääriä. Jatkokehityksessä työkaluun pitäisi lisätä uusia laskettavia arvoja.

7.1.5. Piirtoskriptin generointi

Piirtoskripti generoidaan kaikesta siitä tiedosta, joka on lisätty BuildingChart-olioon. Ohjelmoija pystyy kutsumaan kaikkia BuildingChart-luokan ja perityn Chart-luokan metodeita.

Jos käytetään oletusasetuksia eikä tuloksia rajata, skripti voidaan luoda kutsuamalla kahta funktiota BuildingChartista esimerkiksi `createGroupedBuildingsPerArea`-funktiota ja `drawGraph`-funktiota. Tällöin luodaan ryhmät konstruktorissa syötetyn `group_by`-kentän mukaan. Lopuksi niille lasketaan vuotuiset kesvikulutukset neliötä kohti konstruktorissa syötetyn `compared_field`-arvon perusteella.

7.2. Käyttöliittymä

Raportoinnin käyttöliittymä on toistaiseksi yksinkertainen. Käytössä ei ole vielä mitään tiettyä teemaa, eikä graafiseen ulkoasuun ole kiinnitetty paljoa huomiota. Ylälaidassa on navigointipalkki (ks. kuvio 11), jolla raportoinnissa voidaan siirtyä kuvaajiin, rakennusten vertailuun, palata Energiajunioriin tai kirjautua ulos.



Kuvio 11. Raportointijärjestelmän yläreunan navigointipalkki

Kuvaajat-painike vie käyttäjän järjestelmän kuvaajien piirtosivulle (ks. kuvio 10). Vertailu-painikkeesta käyttäjä pääsee projektien vertailusivulle, jossa käyttäjä voi vertailla kahta tai kolmea omaa projektiaan (ks. kuvio 12). Palaa energiajunioriin -painike vie käyttäjän takaisin Energiajuniorin sivuille. Kirjaudu ulos -painike kirjaa käyttäjän ulos raportoinnista.

Kuvaajat-sivulla käyttäjä voi piirtää erilaisia kuvaajia rakennuksista haluamallaan tiedoilla (ks. kuvio 10). Kuvaajan piirto aloitetaan valitsemalla kuvaajan tyyppi, joka tarkoittaa kuvaajassa esitettävää suuretta. Valittavia vaihtoehtoja ovat lämmitystarve, sähkön kulutus ja kappalemäärä. Tämän jälkeen valitaan ryhmitysperuste, jonka mukaan rakennusten tiedot ryhmitetään, esimerkiksi kuviossa 9 rakennukset on ryhmitetty valmistumisvuoden mukaan. Erilaisia ryhmitysperusteita ovat valmistumisvuosi, pinta-ala, E-luokka, rakennustyyppi, kunta ja lämmitysjärjestelmät.

Piirrettävän datan valinnan jälkeen voidaan lisätä suodattimia, jotka raajaavat mukana olevia rakennuksia. Kuitenkin mukana olevista rakennustyypeistä on valittava ainakin yksi. Lisäksi vähintään yksi lämmitysjärjestelmätyyppi täytyy olla valittuna, jotta on mahdollista löytää rakennuksia.

Lisäksi tuloksia voidaan rajata lisäämällä halutut kunnat, joista rakennuksia haetaan. Rakennusvuoden voi antaa aikavälinä. Rakennus-pudotusvalikolla voidaan valita halutaanko hakea olemassa olevia rakennuksia, uudisrakennuksia vai hakea kaikista rakennuksista.

Ryhmitettäessä rakennuksia valmistumisvuoden tai pinta-alan mukaan voidaan myös määrittää, kuinka moneen ryhmään rakennukset halutaan jakaa. Tämä tapahtuu syöttämällä haluttu lukumäärä ”ryhmien määrä” -kenttään.

Tallenna-painikkeella käyttäjä voi tallentaa tekemänsä haun ja antaa sille nimen. Tallennuksen jälkeen käyttäjä löytää omat tallennetut hakunsa ”Tallennetut suodattimet” -painikkeen alta. Lisäksi käyttäjälle annetaan linkki, joka voidaan jakaa jopa järjestelmään rekisteröitymättömille käyttäjille. Avaamalla tämä linkki päästään katselemaan tehtyä hakua. Haku suoritetaan

aina uudelleen avattaessa, jolloin käytössä on aina uusin data. Jos käyttäjä haluaa tallentaa juuri tietyn ajan hetken tilanteen täytyy tallentaa kuvaaja JPEG- tai PNG-kuvatiedostona paikallisesti kuvaajan oikeasta yläkulmasta löytyvällä painikkeella.

Vertailu

Vertailtavat projektit ?

Hae

Nimi	As Oy <input type="text"/>	As Oy <input type="text"/>	As Oy <input type="text"/>
Osoite	<input type="text"/>	<input type="text"/>	<input type="text"/>
Kunta	Kuhmoinen	Vaasa	Pöytyä
Valitusvuosi	1979	2006	2000
Projektin tyyppi	Pien-, rivi- tai ketjutalo	Pien-, rivi- tai ketjutalo	Pien-, rivi- tai ketjutalo
E-luku	254	310	326
E-luokka	E	E	E
Pinta-ala (m²)	324	346	151
Ilmanvuotoluku	4	4	4
Lämmitystarve (kWh/a)	37801	27776	19900
Tuotettu aurinkoenergia (kWh/a)	0	0	0
Lämpökuormat (kWh/a)	23914	27888	11746
Fossiiliset polttoaineet (kWh/a)	64835	0	0
Uusiutuva energia (kWh/a)	0	0	3333
Lämmitysjärjestelmät	Kulutus (kWh/a)	Kulutus (kWh/a)	Kulutus (kWh/a)
	Standardi öljy: 65156	Suora sähkö: 53662	Takka: 3333
			Suora sähkö: 23904

Kuvio 12. Kuvakaappaus projektien vertailusta

Käyttöliittymä on tehty Bootstrap 3:n päälle. Käyttöliittymän manipulointi on toteutettu jQueryllä. Kuntien valinnassa on käytetty Chosen-hakukenttää, jolla saadaan aikaan valintalaatikko, josta voi hakea ja valita useita vaihtoehtoja (ks. kuvio 10). Lisäksi lämmitysjärjestelmien valinta on toteutettu Bootstrap-multiselectillä (ks. liite 4).

Käyttöliittymä ei anna käyttäjän tehdä valintoja, joilla ei varmasti löydetä rakennuksia. Mikäli käyttäjä yrittää mahdotonta hakua järjestelmä ilmoittaa sivun ylälaitaan ilmestyvällä viestikentällä, missä virhe on.

7.3. Jatkokehitys

Raportointijärjestelmän kaikki suunniteltu toiminnallisuus on valmis, mutta kehitys ei kuitenkaan ole ohi. Ulkoasuun ei ole panostettu vielä juuri ollenkaan, ja

myös esitettäviä arvoja tarvittaisiin lisää. Myös tietoja täytyy lisätä vertailua varten. Raportointi on kuitenkin jo niin pitkällä, ettei siinä ollut yhtään tunnettua bugia työn päättyessä.

Ulkoasu on toistaiseksi Bootstrap 3 vakioteemalla toteutettu. Se kaipaisi jonkinlaista yhteyttä Energiajuniorin vihreään värimaailmaan. Lisäksi osa kappalemäärä-kuvaajista kannattaisi esittää enemmän piirakkagraafeina.

Alueita vertailevan raportin tuottaminen pitäisi lisätä raportointijärjestelmään. Tämä voitaisiin toteuttaa esimerkiksi luomalla toiminto, jossa voidaan valita tallennetut kuvaajat, joista sitten generoidaan raportti.

Vertailu on tällä hetkellä hyvin yksinkertainen, eikä se sisällä paljoa tietoa. Täähän löytyy varmasti lisää esitettävää sisältöä, kunhan järjestelmä otetaan käyttöön.

8. POHDINTA

Työn tavoitteena oli luoda RESCA-hankkeelle raportointijärjestelmä ja tutkia erilaisia web-pohjaisia piirtomenetelmiä. Näistä menetelmistä etsittiin parhaat mahdolliset tietyille käyttötapauksille.

RESCA:n raportointijärjestelmä saatiin kaikkien vaatimusten osalta valmiiksi, vaikka raportoinnin ulkoasua täytyy vielä kohentaa. Käytössä on Bootstrapin va-kioteema, joka on värimaailmaltaan hyvin yksinkertainen. Tästä saattaa aiheutua käyttäjälle käsitys, että systeemi olisi halpa tai jopa mahdollisesti huijausta. Lisäksi vertailussa saatiin valittua myös piirtokirjastot halutuille kolmelle muulle käyttötapaukselle.

Järjestelmästä saadaan tuotettua energiatehokkuutta kuvaavia raportteja tiettyjen alueiden rakennuksista. Alueellista energiatehokkuutta voidaan verrata esimerkiksi kuviossa 10 nähtävällä tavalla vertailemalla kaupunkien energiatehokkuutta lämmitystarpeen kautta. Lisäksi kehitystä voidaan seurata esimerkiksi rakennusvuoden mukaan ryhmittämällä ja vertailemalla saatuja tuloksia. Raportointiin viime hetkellä lisätty rakennusten vertailutoiminto sisältää hyvin vähän tietoja. Näitä vertailtavia tietoja voidaan paremmin lisätä vasta sitten kun järjestelmästä saadaan käyttäjäpalautetta.

Näillä raporteilla voidaan varmasti parantaa rakennusten energiatehokkuutta paikantamalla energiatehokkuudeltaan heikkoja ratkaisuja. Nämä energiatehokkuudeltaan huonot ratkaisut voidaan siten välttää ja puolestaan suosia energiatehokkuudeltaan hyviä ratkaisuja.

Työssä olisi mahdollisesti voitu testata kaiken tyyppisiä kirjastoja, mutta tämä päätettiin jättää pois, koska raportointijärjestelmä vaati JavaScript-kirjaston. Täten ei ollut mielekasta testata esimerkiksi PHP-kirjastoja, joita myös laajalti käytetään.

LÄHTEET

AMcharts. 2014. AMcharts kotisivut. Viitattu 4.8.2014.

[Http://www.amcharts.com/](http://www.amcharts.com/).

Canvas.js. 2014. Canvas.js kotisivut. Viitattu 4.8.2014. [Http://canvasjs.com/](http://canvasjs.com/).

Comparison of JavaScript charting frameworks. 2014. Wikipedian listaus JavaScript-piirtokirjastoista. Viitattu 23.5.2014.

[Http://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_frameworks](http://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_frameworks).

Energiajunior. 2014. Energiajunior etusivu. Viitattu 9.9.2014.

[Https://www.energiajunior.fi/](https://www.energiajunior.fi/).

Google Charts. 2014. Google Charts kotisivut. Viitattu 4.8.2014.

[Https://developers.google.com/chart/](https://developers.google.com/chart/).

Highcharts. 2014. Highchartsin kotisivut. Viitattu 28.7.2014.

[Http://www.highcharts.com/](http://www.highcharts.com/).

Image Processing and GD. 2014. PHP.netin GD kirjaston dokumentaatio. Viitattu 25.5.2014. [Http://fi2.php.net/manual/en/book.image.php](http://fi2.php.net/manual/en/book.image.php).

JpGraph. 2014. JpGraphin kotisivut. Viitattu 11.8.2014. [Http://jpgraph.net/](http://jpgraph.net/).

Pchart. 2014. Pchartin kotisivut. Viitattu 11.8.2014. [Http://www.pchart.net/](http://www.pchart.net/).

PHP/SWF Charts. 2014. PHP/SWF Charts kotisivut. Viitattu 9.9.2014.

[Http://www.maani.us/charts4/](http://www.maani.us/charts4/).

PhpChart. 2014. PhpChartin kotisivut. Viitattu 23.5.2014. [Http://phpchart.org/](http://phpchart.org/).

RESCA Oulu. 2014. RESCA Oulun kotisivut. Viitattu 8.5.2014.

[Http://www.rescaoulu.fi/](http://www.rescaoulu.fi/).

Visifire. 2014. Visifire kotisivut. Viitattu 9.9.2014. [Http://www.visifire.com/](http://www.visifire.com/).

LIITTEET

Liite 1: MySQL koodi, testidatan generointia varten

```
DELIMITER $$
```

```
CREATE PROCEDURE add_test_measurements()
```

```
BEGIN
```

```
    DECLARE i INT DEFAULT 1;
```

```
    WHILE i < 200000 DO
```

```
        INSERT INTO measurement (gauge_id, value, measure_time) VALUES
```

```
            (1316,
```

```
            RAND()+i,
```

```
            DATE_ADD(DATE_ADD(NOW(), INTERVAL FLOOR(RAND()) * 30)
```

```
SECOND), INTERVAL i * 4 MINUTE));
```

```
        SET i = i + 1;
```

```
    END WHILE;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL add_test_measurements();
```

```
DROP PROCEDURE add_test_measurements;
```


Liite 2: Statistikkatauluihin tallennettavat tiedot

Statistikkatiedot

Field	Type	Field	Type
<u>id</u>	mediumint(8)	distr_cooling	float
project_id	mediumint(8)	lights_dev	float
user_id	mediumint(8)	a_limit	smallint(6)
address	varchar(35)	b_limit	smallint(6)
postcode	char(5)	c_limit	smallint(6)
city	varchar(25)	d_limit	smallint(6)
commune_id	smallint(6)	e_limit	smallint(6)
commune	varchar(25)	f_limit	smallint(6)
ready_year	smallint(4)	sun_prod	float
project_type	tinyint(4)	tuotto_solar	float
e_value	smallint(6)	qhhp_jako_y	float
e_class	char(1)	qlt_netto_y	float
area_floor	float	qlt_iv_netto_y	float
iv_lto	float	qlkv_netto_y	float
infiltration_coeff	float	cooling	float
new_old	int(11)	qhsol	float
electricity	float	qh_people	float
fossil_fuel	float	qh_dev	float
renewable_fuel	float	qh_lights	float
distr_heating	float	qh_lkv	float

Lämmitysjärjestelmien statistiikat

Field	Type
<u>id</u>	mediumint(8)
stats_id	mediumint(8)
hs_type	tinyint(4)
tilat	float
tuloilma	float
lkv	float
apulaitteet	float
ostoenergia	float

Liite 3: Piirtoluokkien funktiot

Chart-luokan funktiot

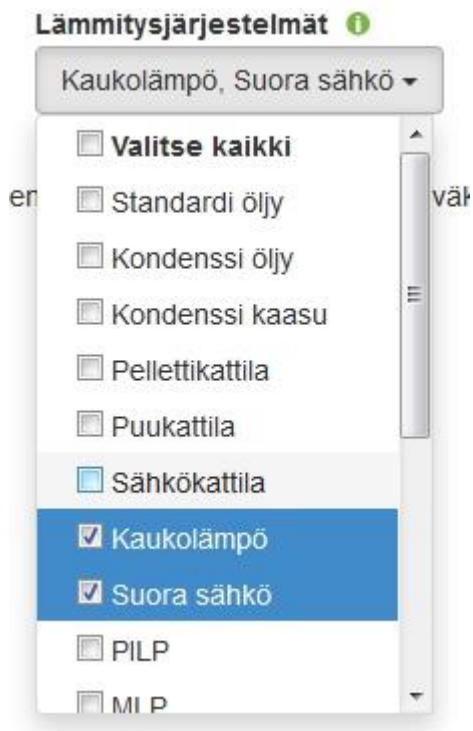
m	🔗	__construct():Raportointi\includes\chart\Chart
m	🔒	createGroups(data : Array, group_by : string, number_of_groups : int, [own_groups : bool = FALSE]):array
m	🔒	calculateGroupAverages(compared_field):array
m	🔒	calculateGroupLimits(group_by : String):Array
m	🔒	createGroupDisplayData(display_values : array, axis_labels : array):void
m	🔒	createDisplayData(data : Array, display_key, label : String, [is_multi_data : bool = FALSE]):void
m	🔒	createData():string
m	🔗	drawGraph([container_id : string = "chartContainer"]):string
m	🔗	enableZoom():void
m	🔗	enableIndexLabels([label : string = "{y}"):void
m	🔗	hideLegend():void
m	🔗	getGroupLimits()
m	🔗	setTitle(title):void
m	🔗	setGraphType(graph_type):void
m	🔗	setAxisTitle(axis_title):void
m	🔗	setLegendText(legend_text):void
m	🔗	setFontSize(font_size):void
m	🔗	setFontColor(color):void
m	🔗	setDataColor(data_color):void
m	🔗	setDataPointColor(data_color : Array):void
m	🔗	setInterval(interval):void
m	🔗	setLabelAngle(label_angle):void
m	🔗	setMarginY(margin_y):void
m	🔗	setMarginX(margin_x):void
m	🔒	setDisplayValues(display_values):void

BuildingChart-luokan funktiot

m	🔗	__construct(group_by, compared_field):Raportointi\includes\chart\BuildingChart
m	🔒	fetchBuildingData():void
m	🔒	generateDisplayValuesPerArea(grouped_buildings):Array
m	🔒	generateDisplayValuesPerItem(grouped_buildings : Array):Array
m	🔒	sortDisplayDataByValue(display_values, labels)
m	🔗	addFilter(column : string, operator : string, value : \Raportointi\includes\chart\Ehdon):void
m	🔗	addAdditionalFilters(column : String, value : String):void
m	🔗	createGroupedBuildingsPerArea([number_of_groups : int number = 10]):bool
m	🔗	createGroupedBuildingsPerItem([number_of_groups : int = 10]):bool
m	🔗	createGroupedBuildingsAverage([number_of_groups : int = 10]):bool
m	🔒	calculateNumberOfBuildingsInGroup(grouped_buildings : Array):void
m	🔒	requiresOwnGroups():bool
m	🔗	getBuildings()
m	🔗	getAxisLabels()
m	🔗	getNumberOfBuildingsInGroup():array
m	🔒	getAdditionalFilters():string
m	🔗	setProjectTypes(project_types : Array):void
m	🔗	setHsFilters(hs_types):void

Liite 4: Bootstrap-multiselect esimerkki

Bootstrap-multiselectillä toteutettu lämmitysjärjestelmien valinta



Liite 5: Piirtoluokkien käytön koodiesimerkki

// Tarkastetaan postin oikeellisuus, ettei sitä ole käpälöity.

```
$privileges->checkPostReport();
```

// Luodaan BuildingChart-olio ja annetaan sille ryhmitysperuste ja kuvaajan tyyppi parametrina

```
$chart = new BuildingChart($_POST['group_by'], $_POST['chart_type']);
```

// Asetetaan valitut lämmitysjärjestelmät filttereihin

```
$chart->setHsFilters($hs_filters);
```

// Asetetaan valitut projektityypit filttereihin

```
$chart->setProjectTypes($project_types);
```

// Lisätään kaikki valitut kunnat

```
foreach($_POST['commune'] as $commune){
```

```
    $chart->addAdditionalFilters('commune', $commune);
```

```
}
```

```
// Rajataan rakennuksia valmistumisvuoden mukaan
if(!empty($_POST['start_year'])) { $chart->addFilter('ready_year', '>=',
$_POST['start_year']); }
if(!empty($_POST['end_year'])) { $chart->addFilter('ready_year', '<=',
$_POST['end_year']); }

$chart->setTitle(createTitle());

if($chart_type == "electricity"){
    $chart->setLegendText("Sähkönkulutus kwh/m2/a");
} else if($chart_type == "kpl"){
    $chart->setLegendText("Kappalemäärä");
} else {
    $chart->setLegendText("Lämmitystarve kwh/m2/a");
}

// Kyseessä kuvaajan tyyppi: kappalemäärä. Luodaan ryhmät
if($chart_type == "kpl"){
    $chart->createGroupedBuildingsPerItem($number_of_groups);
} else {
    // Ryhmitetään rakennukset ja lasketaan arvot neliötä kohti
    $chart->createGroupedBuildingsPerArea($number_of_groups);
}

// Piirretään
echo $chart->drawGraph();
```